

Sequential and non-sequential modes of output

There are two modes of screen display, one sequential the other non-sequential. We have been by now fairly familiar with the former. Programme 1 tests the latter.

Programme 1 *Programme to test non-sequential output*

```

1 /* programme to test non-sequential mode of output, Kit Tyabandha, 14 Dec 2006 */
2 #include< curses.h>
3 #include<stdio.h>
4 int main(){
5     initscr();
6     move(20,20);
7     addstr("test");
8     getch();
9     endwin();
10    return 0;
11 }
```

We next look at how assembly codes are generated from each compilation. Programme 2 gives a simple code. We compile this programme with

```

gcc -c -g -Wall tst.c
gcc -g tst.o -o tst -lcurses -ldl -lm
```

our programme being *tst.c*. Listing 3 shows the Assembly code obtained from the compilation we did above.

Programme 2 *A simple programme*

```

1 /* looking at the assembly codes generated from our compilation,
2    Kit Tyabandha, 18 Dec 2006 */
3 #include<stdio.h>
4 #include<stdlib.h>
5 int main(){
6     int a, b, c;
7     a =1;
8     b =2;
9     c =a+b;
10    printf("\n c = %d\n\n",c);
11    exit(0);
12 }
```

Listing 3 *Assembly code from a compilation*

```
.file "tst.c"
.section .rodata
.LC0:
.string "\n c = %d\n\n"
.text
.globl main
.type main, @function
main:
pushl %ebp
movl %esp, %ebp
subl $24, %esp
andl $-16, %esp
movl $0, %eax
subl %eax, %esp
movl $1, -4(%ebp)
movl $2, -8(%ebp)
movl -8(%ebp), %eax
addl -4(%ebp), %eax
movl %eax, -12(%ebp)
movl -12(%ebp), %eax
movl %eax, 4(%esp)
movl $.LC0, (%esp)
call printf
movl $0, (%esp)
call exit
.size main, .-main
.section .note.GNU-stack,"",@progbits
.ident "GCC: (GNU) 3.3.5 (Debian 1:3.3.5-3)"
```